

Fig. 1

## 

Example Incoming Ethernet Frame With TCP/IP Packet Payload Containing An HTTP Request Message

	Ethernet frame header				Ethernet frame payload (IP packet)												`	
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1						( IP packet	header			·	IP packet payload (TCP packet)							
• • • • • • • • • • • • • • • • • • •		~		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1						*	TCP packet header	,		TCP packet payload	人 (HTTP Request	Message)	/ 0	<b>6</b>
} 8 bytes	] destination physical address – 6 bytes (shown in hex format)	source physical address – 6 bytes (hex format)	} payload type - 2 bytes (hex format - 08-00 is IP packet)		$\}$ payload protocol – I byte (hex format – 06 is TCP)	} 2 bytes	source IP address – 4 bytes (byte-decimal format)	} destination IP address – 4 bytes (byte-decimal format)	} variable length	source port - 2 bytes (decimal format)	$\}$ destination port – 2 bytes (decimal format – 80 is HTTP)	3 20 bytes (hex format)	} HTTP request line – variable length (text format)	FI 3 F - T - GUADAI	$\int 1111$ message neaver fields – variable length (text format)		HTTP message body - variable length (text format)	} 4 bytes
<pre><pre><pre></pre></pre></pre>	08-00-20-03-00-50	08-00-20-05-1A-11	00-80	<various fields="" header="" ip=""></various>	90	<ip checksum="" header=""></ip>	153.83.28.125	128.32.78.105	<pre><p header="" options=""></p></pre>	3183	80	<various fields="" header="" tcp=""> 3 20 bytes (hex format)</various>		Host: www.XYZ.com	User-agent: Mozilla/3.04		•••	<frame check="" sequence=""/>
202	204	206	208	210	212	214	216	218	220	222	224	226	228	230	232	234	236	238

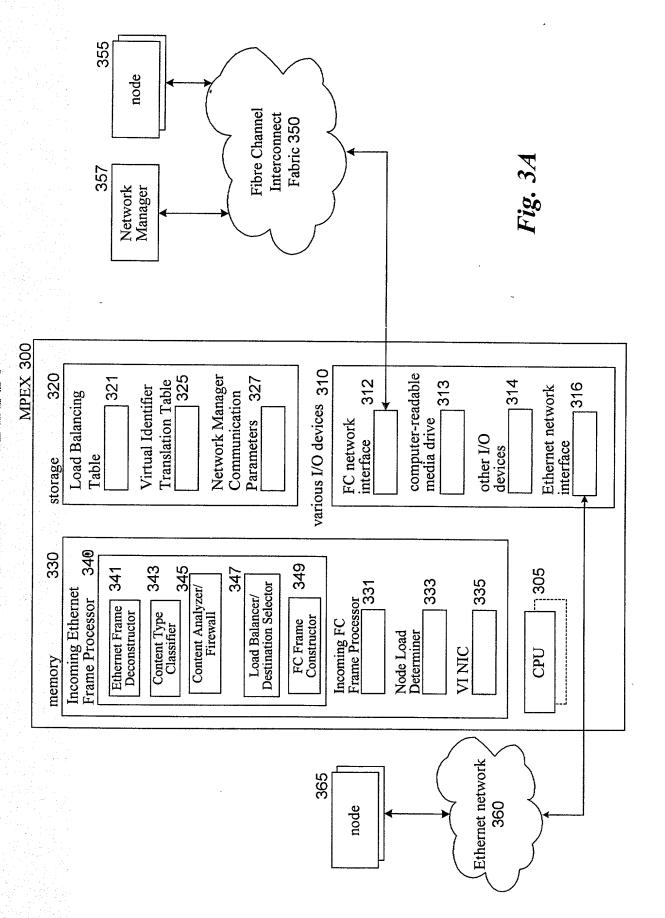
Figure 2A

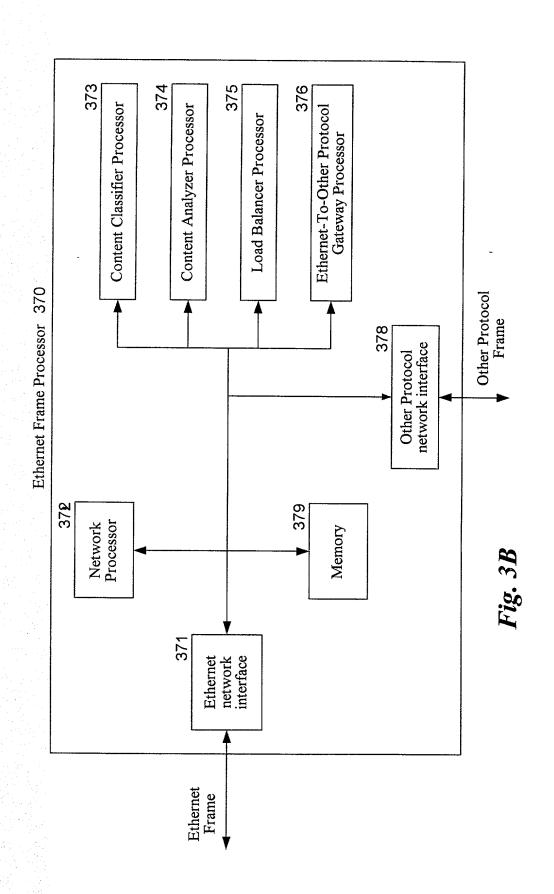
## \*OBSSTAL \*CESSA

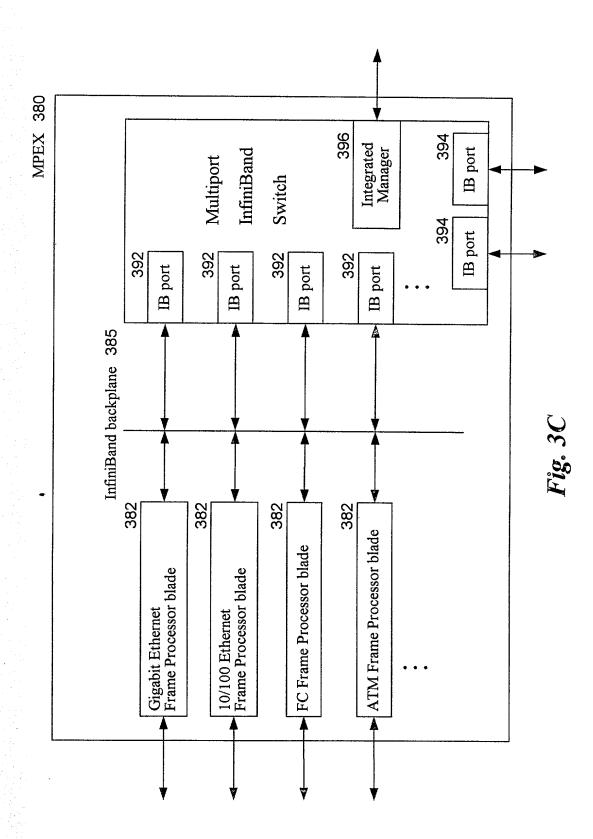
Example Outgoing Fibre Channel Frame With TCP/IP Packet Payload Containing An HTTP Request Message

Fibre Channel	Fibre Channel frame payload (IP packet)									
st)	IP packet header	IP packet  payload  (TCP packet)	3 6 1 1 5 6 6 8 8 8 8 8 8							
(e.g., unsolicited requ e		TCP packet header  TCP packet payload  (HTTP Request  Message)	**************************************							
4 bytes  I byte – specifies frame category (e.g., data) & possibly function (e.g., unsolicited request)  destination port physical address – 3 bytes (shown in hex format)  I byte (can include priority and preemption information)  source port physical address – 3 bytes (hex format)  source port physical address – 3 bytes (hex format)  payload type – 1 byte (hex format – 05 is IP packet)  15 bytes (includes an indication if payload has optional headers)	<pre> } variable length (protocol-specific headers)  } 9 bytes (includes Type of Service and Length)  } payload protocol - 1 byte (hex format 06 is TCP)  } 2 bytes  } source IP address 4 bytes (byte-decimal format)  } destination IP address 4 bytes (byte-decimal format)  } variable length </pre>	<pre>} source port - 2 bytes (decimal format) } destination port - 2 bytes (decimal format - 80 is HTTP) } 20 bytes (hex format) } HTTP request line - variable length (text format) } HTTP message header fields - variable length (text format) } HTTP message body - variable length (text format)</pre>	} 4 bytes } 4 bytes							
<pre><start frame="" of=""></start></pre>	<pre><optional headers=""> <various fields="" header="" ip=""></various></optional></pre>	3183 80  GET /pub/text.html HTTP/1.1 Host: www.XYZ.com User-agent: Mozilla/3.04 :	<cyclic check="" redundancy=""> <end frame="" of=""></end></cyclic>							
252 254 256 256 260 262 262	266 268 270 272 274 276 276	280 282 284 286 286 290 292	296							

Figure 2B







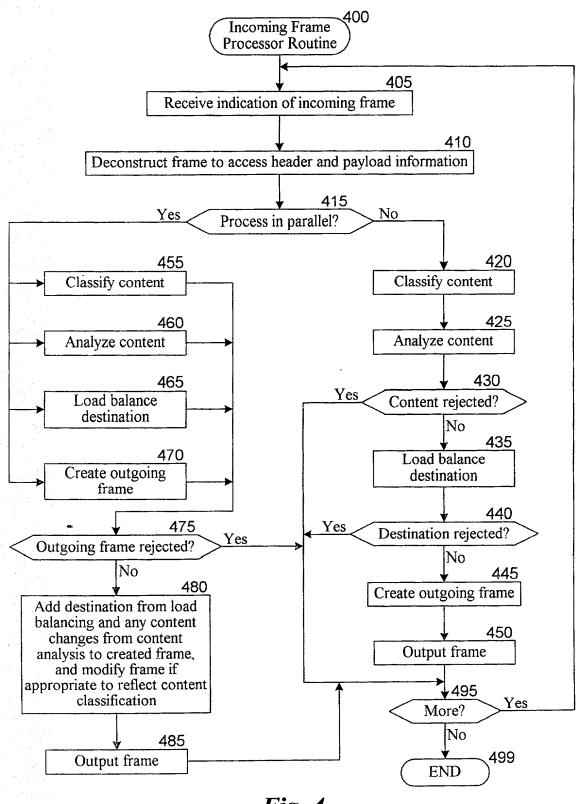


Fig. 4